

## Angular JS and node JS

- **Introduction**
  - Introduction to AngularJS MVC Architecture
  - Conceptual Overview
  - Setting up the Environment
  - First Application
  - Understanding ng Attributes
- **Filters**
  - Purpose of Filters
  - Built-In Filters
  - Uppercase and Lowercase Filters
  - Currency and NumberFormatting Filters
  - OrderBy Filter
  - Filter Filter
  - Creating Custom Filter
- **Expression and data binding**
  - Number and String Expressions
  - Object Binding and Expressions
  - Working with Arrays
  - Forgiving Behaviour
  - Understanding Data Binding
- **Forms**
  - Using Simple Form
  - Working with Select and Options
  - Input Validations
  - Using CSS classes
  - Form Events
  - Custom Model update triggers
  - Custom Validations
- **Routing**
  - Introduction to SPA
  - Creating HTML Templates
  - Configuring Route Provider

- **Controllers**
  - Understanding Controllers
  - Programming Controllers and \$scope object
  - Adding Behaviour to a Scope Object
  - Passing Parameters to the Methods
  - Having Array as members in Controller Scope
  - Nested Controllers and Scope Inheritance
  - Multiple Controllers and their scope
- **Working with directives**
  - Introduction
  - Core Directives
  - Conditional Directives
  - Style Directives
  - Mouse and Keyboard Events Directives
- **Modules**
  - Why Module?
  - Module Loading and Dependencies
  - Recommended Setup of Application
  - Creation vs Retrieval
- **Services**
  - Understanding Services
  - Developing Creating Services
  - Using a Service
  - Injecting Dependencies in a Service

## Node JS

- **Introduction**
  - what is NodeJS and why
  - Advantages and dis
  - Create a simple node application
  - The NodeJS Terminal
  - NodeJS package manager NPM
  - Creating a package for managing your application
- **Node Modules**
  - What is a module
  - Require and module exports
  - Using core modules
- **File System**
  - Node apps and managing a file system
  - Creating and deleting files and directories

- **HTTP and HTTPs**
  - Making a simple server
  - When to use HTTP and HTTPs
  - Server ports and listening
  - HTTP requests and responses
  - Request and response headers and body
  - Creating a response to incoming requests
  - Building a simple HTTP server with static files
- **Express**
  - What is the express framework
  - How it differs to using HTTP and other core modules
  - Creating an express application
- **Express Server**
  - Making an HTTP internal server
  - Attaching express to an external server
  - The role of semantic requests
  - USE, GET, POST, PUT, DELETE, etc.
- **Advance NodeJS**
  - Socket programming
  - Real-time communication with node
  - Accessing nodejs with database
  - Creating chat server with nodejs
- **RESTful APIs**
  - What does RESTful mean?
  - Manipulating data via a route
  - Typical semantic requests for routing an API
  - Persisting data through JSON
  - The role of database technology